

**Algorithms, Data Structures & Complexity**  
***Project: Path-finding in a Grid-map***

Due on last lab session for your group

*Federico Pecora, Uwe Köckemann*

**Uwe Köckemann**

### General Hints

- Some of these exercises require to identify how to represent the described problem as a graph and to identify the correct algorithm to solve the problem.
- All required algorithms have been covered in the lecture.

### Handing In

The project should be completed and shown during the last lab session for your group. The TA will pass by your seat and evaluate each exercise. Upon successful completion of the project, please provide a text file named `project.txt` with the following content:

- names of group members (up to two members per group);
- a short summary of the contribution of each group member;
- indicate which file(s) implement the algorithm and/or data structure you have implemented;
- a brief explanation of the tests that were carried out to test the implementation;
- instructions on how to execute a test to verify the implemented code;

Please submit all lab material collected into an archive (zip, rar, or tar.gz) via a Blackboard message to Uwe Köckemann and Federico Pecora.

**Note:** the project can be done individually or in pairs. Larger groups are *not* allowed. All incidents of plagiarism will be reported. Please write your names on all material you hand in.

### Description

You are working for a company that develops video games. In your current project, non-player characters often need to move between different points in a map. Consider a map as drawn below. Each cell in this map is either white (accessible) or black (unaccessible). As an input to your program you are given a file containing only 0s and 1s (for accessible and unaccessible territory). Implement a simple path finding algorithm to find the shortest paths between accessible points the map loaded from the given file. Consider using one of the data structures implemented in Lab 4 to represent the map as a graph. Diagonal moves are not allowed.

1. Load a map from a file
2. Translate the map into graph
3. Given 2 accessible points in the map find a path if one exists or return failure (NULL)

The input to your program should be 5 arguments:

1. Filename of the map
2.  $x_1$  x-coordinate of the origin point
3.  $y_1$  y-coordinate of the origin point
4.  $x_2$  x-coordinate of the goal point
5.  $y_2$  y-coordinate of the goal point

If a path exists, the output of your program should be a list of coordinates, e.g.,

$(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow \dots$

If no path exists the output should be:

NULL

y ↓    x →	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										